Software Engineering Institute

Software-Intensive Systems Producibility: A Vision and Roadmap (v 0.1)

Grady Campbell

December 2007

TECHNICAL NOTE
CMU/SEI-2007-TN-017

Acquisition Support Program

Carnegie Mellon

This work is sponsored by the U.S. Department of Defense.

The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Table of Contents

1 Conceiving a Software-Intensive Systems Producibility Initiative 1

1.1 Purpose of the Roadmap 1

1.1.1 Terminology 1

1.1.2 Elaboration 3

1.2 Objective and Scope for an SISPI 4

1.2.1 Developer Productivity 4

1.2.2 Product Value 5

1.2.3 Acquirer Acuity 5

1.3 A Reference Producibility Vision 5

1.4 A Normative Technology Advancement Life cycle 7

1.5 Programmatic Approach 8

1.5.1 Precepts 8

1.5.2 Principles 8

1.5.3 Technology Phases 9

1.5.4 Programmatic Risks and Mitigations 10

1.6 Programmatic Context 11

1.6.1 SISPI Summary Business Case 11

1.6.2 SISPI Critical Success Factors for Acquisition and Sustainment 12

1.6.3 Rationale for Producibility-Motivated Change 13

1.6.4 Unrealized Advances 16

1.6.5 Related Efforts 16

2 A Framework for SISPI Research 17

2.1 Model-based Development 17

2.1.1 Definitions 18

2.1.2 Objectives 18

2.1.3 Topics 18

2.1.4 Notional Milestones 19

2.1.5 Elaboration 20

2.1.6 References 23

2.2 Predictable Software Attributes 23

2.2.1 Definitions 23

2.2.2 Objectives 23

2.2.3 Topics 23

2.2.4 Notional Milestones 24

2.2.5 Elaboration 25

2.2.6 References 26

2.3 System Virtualization 26

2.3.1 Definitions 26

2.3.2 Objectives 26

2.3.3 Topics 26

2.3.4 Notional Milestones 27

2.3.5 Elaboration 27

2.3.6 References 28

2.4 Disciplined Methods 28

2.4.1 Definitions 28

2.4.2 Objectives 28

2.4.3 Topics 28

2.4.4 Notional Milestones 29

2.4.5 Elaboration 30
2.4.6 References 32
2.5 Infrastructure and Emerging Technology 32
2.5.1 Definitions 32
2.5.2 Objectives 32
2.5.3 Topics 32
2.5.4 Notional Milestones 32
2.5.5 Elaboration 33
2.5.6 References 34

3 A Framework for SISPI Transition 35
3.1 Validation, Integration, and Productization 36
3.1.1 Definitions 36
3.1.2 Objectives 36
3.1.3 Topics 37
3.1.4 Notional Milestones 37
3.1.5 Elaboration 37
3.1.6 References 39
3.2 Adoption 39
3.2.1 Definitions 39
3.2.2 Objectives 39
3.2.3 Topics 39
3.2.4 Notional Milestones 39
3.2.5 Elaboration 40
3.2.6 References 40

4 Managing SISPI Efforts 41
4.1 Definitions 41
4.2 Objectives 41
4.3 Topics 41
4.4 Notional Milestones 41
4.5 Elaboration 42
4.5.1 Setting Measurable Goals for Research Efforts 42
4.5.2 Selecting SISPI Efforts 42
4.5.3 Opportunities for Near-Term Progress 43
4.5.4 Measuring SISPI Progress 44

Bibliography 49

List of Figures

Figure 1: A Notional SISPI CAD/CAM Process 6

Preface and Acknowledgments

This document is the result of an effort to formulate a strategy for improving the effectiveness of technology used in the development of software-intensive systems. As a result of contributions by numerous industry and academic researchers to workshops sponsored by the Deputy Under Secretary for Defense Science and Technology and surveys of past studies and ongoing research and technology efforts, this report is an initial draft of a framework for a coordinated government-industry effort to focus research and transition activities toward the achievement of enhanced “producibility.” Further work on this framework will focus on refining and elaborating the goals of identified research and transition themes. This framework can then be used as a shared guide for focusing and evaluating the value of proposed research and transition efforts.

Dr. Andre van Tilborg and Rob Gold of OSD have made this work possible. The following people have made many valuable comments and suggestions toward improving the quality of this report:

- Rob Gold (OUSD DDR&E/S&T)
- Ralph Wachter (Office of Naval Research), Jim Kirby and Judy Froscher (Naval Research Laboratory)
- Steven Drager, William McKeever, and Dr. David Luginbuhl (Air Force Research Laboratory)
- Jeff Dehart and Glenn Racine (Army Research Laboratory)
- Dr. Richard Turner (Systems and Software Consortium Inc.)
- Thomas McGibbon (DoD Data and Analysis Center for Software)

A number of other people have given valuable comments on at least one version of this report. Directions being taken by research efforts at Lockheed-Martin, BAE Systems, and Boeing have informed the perspectives proposed in this report. Dr. John Goodenough provided helpful perspectives on producibility aspects identified in the SEI’s Ultra-Large-Scale Systems study.

Abstract

A Software-Intensive Systems Producibility Initiative (http://www.sei.cmu.edu/sispi) has been proposed to foster a program of technology research and transition that will improve producibility in the acquisition/development and sustainment/evolution of software-intensive systems (SiS).

This document is a *draft in progress* of a technology vision and roadmap to improve the ability of the DoD and industry to deliver needed SiS capability in a timely, cost-effective, and predictable manner. The goal at this stage is to establish the general concepts and approach for a producibility initiative and to stimulate discussion of these ideas and the research and transition efforts needed to achieve enhanced producibility in practice.

The roadmap is meant to serve as a coherent evolving framework for defining and prioritizing potential research investments and technology transition efforts related to producibility. A roadmap has three elements: a representation of the current situation, a vision that characterizes an improved situation, and a plan of action for transitioning from the current to the improved situation. This roadmap identifies five research themes, two transition themes, and an approach to measuring effectiveness for an initiative focused on achieving a vision of enhanced SiS producibility.

1 Conceiving a Software-Intensive Systems Producibility Initiative

A Software-Intensive Systems Producibility Initiative (SISPI) is proposed to foster a program of technology research and transition that will improve *producibility* in the acquisition/development and sustainment/evolution of software-intensive systems (SiS). The SISPI is envisioned as a collaboratively funded and directed effort of the U.S. Department of Defense (DoD) and industry for engagement and coordination of a broad spectrum of university and industrial researchers, software tool vendors and transition agents, and SiS user and support communities. Further, the SISPI will coordinate activities as feasible with other U.S. and international efforts pursuing similar objectives.

1.1 PURPOSE OF THE ROADMAP

The purpose of this technology roadmap is to formulate a coherent framework for prioritizing potential research investments and technology transition efforts. The goal of this work is to foster the systematic, progressive advancement of the methods and tools used in producing SiS products, to the benefit of U.S. government and industry. The general references in the bibliography identify principal influences on the directions proposed here.

SISPI governance will establish criteria for judging the priority of proposed research and transition efforts. The roadmap simply provides a systematic framework for understanding the relevance, dependencies, and potential contribution of candidate efforts.

The roadmap is organized into four sections:

- Conception—The context, motivation, vision, and approach for an SISPI
- Research themes—Research advances needed to achieve the SISPI vision
- Transition themes—Transition actions needed to move research advances into practice
- Managing progress—How the SISPI will select its efforts and measure progress

1.1.1 Terminology

The following is a limited set of general terms that provide the basis for discussing the scope and approach of the SISPI. Other relevant terms are defined in subsequent sections as needed depending on their particular usage there.

Acquisition	The enterprise of obtaining and deploying products in support of systems relevant to the mission and enabling operations of a customer
Behavior	The externally observable properties and effects of a system or element
Capability	The means by which a purpose can be achieved

Customer	The people/organization for which a product is acquired and whose objectives and operations determine the criteria by which the acceptability of the product is to be judged
Development	The enterprise of constructing or modifying a product, independently or as a component of an acquisition effort
Domain	The knowledge represented by a family of similar problem-solutions and the expertise required to create corresponding products
Ecosystem	A set of systems whose behaviors interact with and within an environment
Evolution	The enterprise of modifying or replacing a product due to changing needs or technology
Mission	The purpose for which an organization exists
Needs	The considerations that customers identify as desired capabilities, perceived weaknesses, or desired improvements in a system of interest
Objectives	The envisioned end results that lead an organization to undertake particular efforts/actions; the criteria that an enterprise sets for judging its success
Problem	The gap between a system as it exists and the system as would better enable a customer in achieving objectives
Problem-Solution	A unified formulation of a problem and associated satisficing solution (a solution that meets or exceeds the minimum criteria determined by a product's requirements)
Producibility	The ability to deliver needed capability in a timely, cost-effective, and predictable manner
Product	An integrated set of artifacts (work-products) that describe and reduce-to-practice understanding of a problem-solution, for the purpose of transforming a system
Requirements	The criteria, consistent with needs and constraints, that determine whether a product is acceptable as a solution to a problem
SiS	*Software-intensive system*, a system in which a significant degree of essential behavior is realized through software
Solution	A means of transforming a system to resolve an identified problem

Sustainment	The enterprise of maintaining (monitoring, adjusting, repairing) a product that a customer organization has instituted into operational use
System	The processes, conditions, and behaviors that arise due to interactions among a set of constituent elements (devices, people, and other systems) operating within a shared environment
User	Any person or organization that has a role in the operations of an SiS

1.1.2 Elaboration

This document is a “draft in progress” of a roadmap for technology that will improve SiS producibility for DoD and industry. The goal at this stage is to establish the general concepts and approach of a producibility initiative and to stimulate discussion of these ideas and the research and transition efforts needed to achieve enhanced producibility. The details, particularly the suggested notional milestones of the various research and transition themes, are highly susceptible to change for the foreseeable future as comments and suggestions on these issues provide improved insight into what is needed.

Although developers commonly refer to the product of development as being a “system,” “system” in the roadmap refers specifically to the aggregation of elements (devices and users) that constitute a system and to the totality of operational behavior that results from actions of and interactions among these elements. The scope and composition of a system of interest is subjective, determined in relation to the mission and objectives of a concerned organization. The term “product” refers more narrowly to the mechanisms and artifacts that are constructed to express a problem-solution; by injecting a product into a system, we induce changes in the properties and behavior of the system with the aim of enhancing satisfaction of a customer’s objectives. A product encompasses the means to add, remove, or modify devices and to modify the prescribed ways in which people are to operate as elements of the system.

“Enterprise” and “organization” are used with conventional meanings but are mutually defining terms. An enterprise is chartered by an organization with the aim of achieving particular objectives. The operation of the enterprise is, in turn, realized as a responsible (real or virtual) organization. “Objectives” and “capability” are defined in a similarly related manner. Particular objectives lead an organization to acquire/develop capabilities and an organization’s capabilities lead it to take on particular objectives. The interplay of objectives and capabilities determine the essential nature of an organization and its enterprises.

Reflecting the primary focus of this roadmap on the DoD enterprise and the operation of its performing organizations, there are associated endeavors for the realization of SiS through the acquisition/development and sustainment/evolution of products. A distinction is drawn between the “needs” of a customer for new/modified capabilities (to be realized in a product) and the “requirements” that describe the criteria against which the product is judged as acceptable for satisfying those needs. From this perspective, needs may at an arbitrary point be “fixed” (inclusive, however, of anticipated changes) whereas requirements may constantly change as problem-solution understanding evolves. A product is considered “complete” when it can be shown to con-

form to specific requirements. A product is “acceptable” for use if its requirements define a product that satisfies the customer’s needs.

1.2 OBJECTIVE AND SCOPE FOR AN SISPI

The scope of the SISPI derives from the definition given for producibility. The objective of SISPI is to improve our ability to deliver needed capability in a timely, cost-effective, and predictable manner into the operation of software-intensive systems. This ability has three dimensions:

- ***developer productivity***—the efficiency and effectiveness of developers in creating and evolving a product
- ***product value***—the utility and quality of each product that results
- ***acquirer acuity***—the insight and foresight that acquirers have in delineating current and future capabilities needed

To be successful, SISPI must improve each of these and do so in a way that strengthens interdependencies among them.

Producibility concerns technologies of production as opposed to product technologies. Technologies that are used in products are outside the scope of the SISPI. SISPI will exploit product technology advances when applicable to production needs but will focus research efforts on technologies that uniquely benefit production and would not otherwise be addressed. (Examples of technology areas excluded by this are: computing infrastructure software [operating systems, middleware, services], application architectural models, computational algorithms, device drivers, reusable [generic or domain-specific] application software, communications protocols, autonomous agent/robotics software, human-machine interfaces, social/organizational aspects of SiS.)

1.2.1 Developer Productivity

Developer productivity concerns the engineering and management practices that determine the efficiency and effectiveness of the development enterprise. These practices encompass systems, software, and specialty-engineering efforts required in the development of an SiS product, as influenced by

- having relevant domain knowledge and expertise for effective inter-discipline communication and problem-solution resolution
- applying engineering discipline in the use of effective methods within a well-conceived process capability
- utilizing a domain- and process-compatible, effort-reducing technology base
- exploiting existing leveragable (legacy, COTS, open source, domain-specific) assets in creating product content
- identifying and accommodating uncertainty, diversity, and potential change for evolvable products

1.2.2 Product Value

Product value concerns the utility of a product for its intended use and its quality as it affects that use. Producibility focuses on exposing and improving the means by which product value can be systematically attained:

- creating functionality that is cost-effectively responsive to business/mission needs
- exposing and controlling product attributes that determine properties of interest in the desired behavior of a system
- ensuring product compatibility with targeted system and ecosystem operational environments, including customer organization policies and procedures

1.2.3 Acquirer Acuity

Acquirer acuity concerns the acquirer's degree of insight about current customer needs and foresight about future needs, in order to determine a cost-effective means of providing the customer with a satisfactory product. Proper acuity requires both acquirer expertise and sound acquisition practices:

- applying producibility-enabling acquisition policies and practices
- properly communicating a customer's current and changing needs in light of operational context and uncertainties across an SiS product life cycle
- providing effective technical direction, oversight, and feedback, utilizing mechanisms that ensure capability-cost-schedule predictability and resolution of tradeoffs
- supplying adequate infrastructure for technology development, evaluation, transition-into-use, and evolution

1.3 A REFERENCE PRODUCIBILITY VISION

A roadmap has three elements: a representation of the current situation, a vision that characterizes an improved situation, and a plan of action for transitioning from the current to the improved situation. This section proposes a vision that motivates and focuses the described SISPI:

CAD/CAM for Software-intensive Systems

CAD/CAM (computer-aided design and manufacturing) is referenced here in its broadest sense, CAD being the conception, design, and analysis of a problem-solution in model form and CAM being the manufacture from raw and processed materials of a product that conforms to the problem-solution model. Realizations of the CAD/CAM concept are common today, with varying degrees of completeness, in most manufacturing industries. While all of the elements exist today for rudimentary forms of SiS CAD/CAM, there are few and only limited examples of this approach in real-world practice.

Figure 1 (from a work describing an advanced product line development approach [O'Connor 1995]) notionally depicts an equivalent SiS CAD/CAM capability. The key elements in this depiction are the Application Model through which a problem-solution is expressed and analyzed and the Application Product which is mechanically derived from the Application Model using formalized domain knowledge.

The purpose of this vision for the SISPI is to motivate a reconception of how software-intensive systems are created and evolved. The traditional conception is based on the progressive construction of unique constituent parts and their integration into a functioning whole. This new conception views a system as an organic whole that must be iteratively refined to exhibit desired behavior. The behavior of a system is modified through the application of engineered products that override particular aspects of its previous behavior. Those products derive mechanically from models that describe desired behavior and support analyses that provide insight into its implications for the system.

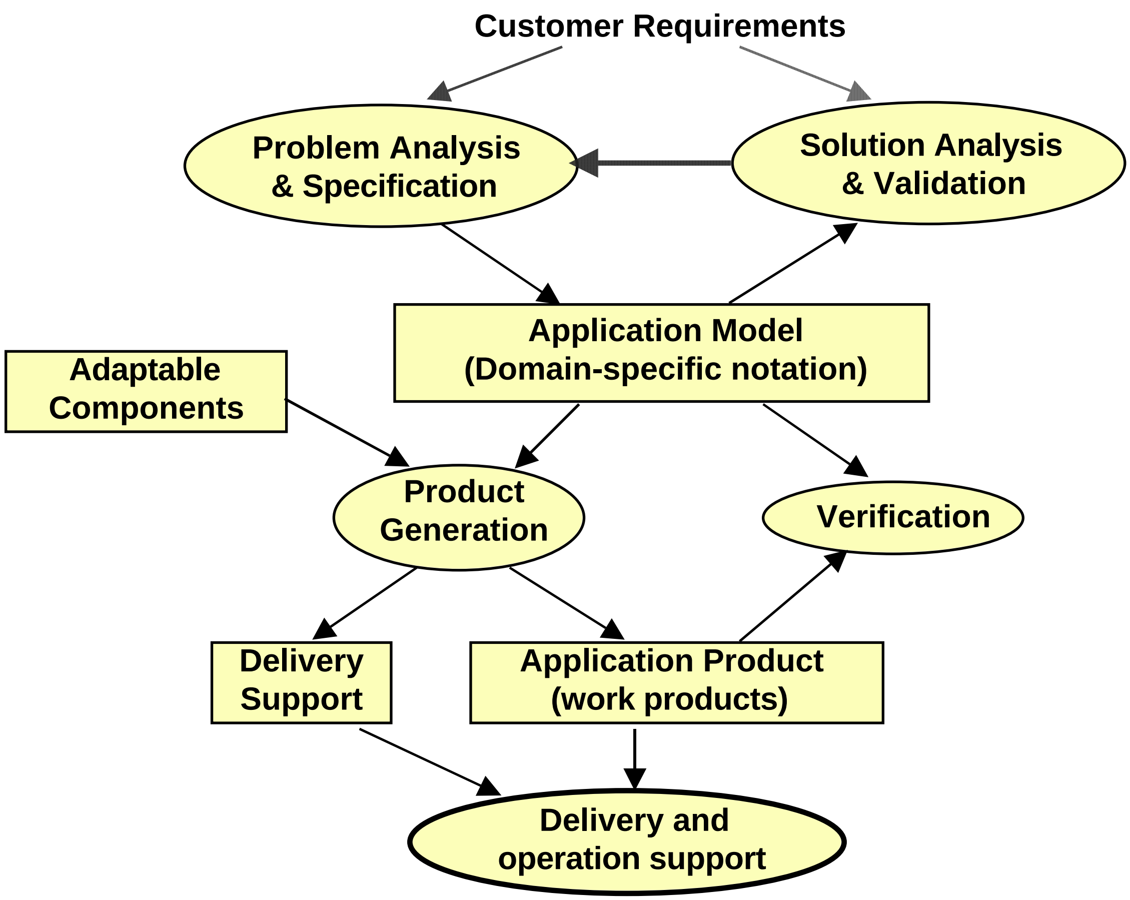

Figure 1: A Notional SISPI CAD/CAM Process

Five principles characterize this vision:

- model-centric—All problem-solution information is expressed in a comprehensive multi-faceted model of a product and its envisioned context of use.
- virtualized—A system is defined by building, pre-deploying, and validating in a software form within a hardware/software/user virtual environment.
- predictable—Software and dependent system properties of interest are able to be accurately predicted and mutually optimized as a product model evolves.

- decision-focused—Multiple alternative solutions are modeled, produced, and empirically evaluated based on identified customer and engineering decisions.
- evolvable—The problem-solution is continuously evolved to create variant products that satisfy anticipated differing or changing needs.

1.4 A NORMATIVE TECHNOLOGY ADVANCEMENT LIFE CYCLE

Systematically advancing SiS producibility practices is not simple and is unlikely to occur quickly. As a basis for developing a roadmap, there are broadly four stages in taking technology from concept to practice. Advancing producibility will require effective coordinated performance of all of these activities for a broad vision-derived collection of technologies targeted by the SISPI.

<u>Research—Define the technology</u>

1. Descriptive method defined
2. Prescriptive/practicable guidance documented, with customization options
3. Prototype tool supporting method implemented
4. Training materials developed

<u>Validation—Evaluate the technology</u>

5. Adoption criteria and guidance provided
6. Applicability of the technology demonstrated on representative problems
7. Cost-benefit of the technology in use estimated

<u>Integration and Productization—Prepare the technology for use</u>

8. Incompatibilities in data/procedures between the technology and current/related tools and practices resolved to enable consistent usage
9. Prototyped technology productized in accordance with sound engineering criteria to enable efficient and reliable operation
10. Productized tools and methods integrated as product offerings of technology providers
11. Support services/resources (mentoring, help, training) established to assist adopters in effective adoption and use of the technology

<u>Adoption—Institute use of the technology</u>

12. Transition agents and activities chartered to assist each potential adopter (acquisition program or industry) in evaluating, preparing to use, and instituting appropriate technology advances
13. Texts and courses that explain the technology and its use developed and delivered by educators
14. Case studies written showing experiences, results, and lessons of using the technology
15. Economic analyses of technology uses developed to provide evidence of relative viability

1.5 PROGRAMMATIC APPROACH

To be effective, the SISPI program must repeatedly answer three questions:

1. What is the current context, the state of Producibility as it relates to DoD Acquisition, and what technology advances would improve this?
2. Given the current context, what must be done to transition technology improvements into common practice?
3. Given SISPI efforts to advance producibility, are anticipated improvements in DoD acquisition being achieved?

These questions correspond to (1) establishing a **research** agenda for advancing producibility, (2) orchestrating the **transition** of technology advances into use, and (3) achieving **measurable** improvements in SiS acquisition traceable to SISPI efforts. These questions dictate the ongoing operation of the program and the answers to these should be rigorously reviewed annually to ensure that the program is operating properly and effectively in keeping with its objectives.

1.5.1 Precepts

The traditional approach to acquisition and systems/software engineering is a poor fit, both in theory and as practiced, for building complex software-intensive systems.

- It formulates needs in ad hoc forms that are often poorly organized, incomplete, and either ambiguous or overly specific.
- It tends to focus in excessive detail on describing ideal behavior but neglects to define behavior under resource-constrained or failure conditions.
- It fails to expose and bridge differing concepts and assumptions that prevent effective communication among domain experts, systems engineers, software engineers, and acquirers.
- It suppresses awareness of uncertainty and potential changes in needs as significant sources of insight to developers.
- It fails to expose and manage the codependence between systems and software engineering, including implications of requirements and systems engineering decisions on software and effects of software decisions on system capabilities.
- It has an over-reliance on testing as the primary means of evaluating acceptable software capability and quality.
- It has an over-dependence on secondary sources (documentation, expert opinion) versus direct evidence from use as the means of monitoring progress.

1.5.2 Principles

Traditional approaches are flawed in part because they make simplifying assumptions about the needs that motivate SiS products. These approaches assume a simplified linear process that does not reflect how SiS developers actually work. An improved approach must better reflect what SiS customers and developers need to work effectively:

- complexity—SiS needs are inherently complex to satisfy and grow more complex faster than our capabilities to satisfy them increase. Approaches that depend on unaided human comprehension working at low levels of detail overwhelm our abilities. We need the means to work

at higher conceptual levels without losing our intuitive grasp of essential details and the implications of choices to be made.

- constraints—Resources for creating SiS products and for making best use of them are inevitably limited. We must have the means both to enhance our productive capabilities and to resolve tradeoffs that arise due to technical, physical, economic, and social constraints on development efforts.
- uncertainty—The challenges faced by organizations change over time, so that problems and needed solutions change. Such change introduces uncertainty into our ability not only to create a solution but even to fully understand the problem. Even when needs are not changing, our imperfect understanding is a source of uncertainty in our ability to create an acceptable product. The certainty of uncertainty means that we must be able to build products that are continually changeable as either needs or understanding change.
- diversity—The perfect product is one that conforms to an organization's business needs and its ways of doing business. Despite similarities, organizations' needs differ and the products they use should be customized to best serve those needs. Similarly, there is no single process/method-set that is best for building a product in any organization and any technical or business domain; at the least, there is the need for customization to reflect differing tradeoffs related to enabling technology or business constraints.
- commonality—Common needs exist across programs, for technology that provides needed infrastructure for a system or is otherwise not subject-matter sensitive and for the application of subject matter that is similarly relevant in different business area domains.
- iterative refinement—Few human products are perfect, or even "good enough," without efforts to improve them; iterative, usage-based improvement is essential for success in that good solutions must properly balance tradeoffs among competing concerns that often are fully understood only as a result of product use.

1.5.3 Technology Phases

SISPI efforts are organized to reflect the phases of the technology advancement life cycle above:

- research—Enable producibility advances by creating improved techniques for software/systems acquisition, development (management and engineering), and sustainment.
- transition—Transform industry practice for improved producibility based on results from research.
  - validation: Demonstrate the applicability and practical value of research results for building DoD systems.
  - integration and productization: Engineer research results into integrated engineering tools and methods suitable for adoption and production use.
  - adoption: Facilitate the adoption by industry and DoD programs of appropriate producibility-enhancing tools and methods.

1.5.4 Programmatic Risks and Mitigations

A programmatic risk is any circumstance that could impede the success of the SISPI. For each risk, a mitigation is an action that the SISPI can take to reduce the likelihood or impact of that risk.

1. Sound producibility technologies have transition costs that are not acceptable to permit acquisition programs to adopt and institutionalize them:
   - Give priority to technologies that can be encompassed or easily integrated with tools currently used by programs.
   - Require research and transition efforts to address adoptability (ease of learning and use) of technologies by acquisition program (government and industry) users.
2. Commercial software tool-method vendors are unwilling to evolve or supplant existing capabilities to accommodate producibility technologies:
   - Give priority to technologies that vendors judge as marketable and transitionable to their customers.
   - Sponsor open source or small business efforts to create new technology tooling.
   - Negotiate commitments from acquisition/industry organizations to purchase tools using needed technologies to reduce vendor investment risk.
   - Advocate and assist programs to create specialized tooling that meets their particular needs.
3. Producibility improvements require changing too much at once, resulting in a perception of excessive cost or risk for potential adopters:
   - Identify improvements that require only well-contained changes within current practice.
   - Provide funded resources that minimize the time, costs, and learning curve for an adopting program.
   - Require research efforts to adequately characterize assumptions about context and dependencies of using technologies and to adjust these to match acquisition program realities.
   - Give priority to research efforts that can provide alternative manual or conventionally tool-assisted methods in lieu of requiring use of new and unproven or costly tooling.
4. Proposed research fails to address identified needs or fails to demonstrate anticipated benefits within a needed timeframe:
   - Base research funding on adherence to priorities and dependencies identified in the roadmap.
   - Require research efforts to identify frequent milestones that demonstrate the degree to which expected benefits are feasible in acquisition use.
   - Require research efforts to clearly identify and plan around unknowns and alternatives to ensure delivery of a best-available result for timely transition into use.

1.6 PROGRAMMATIC CONTEXT

The context for SISPI actions is (1) DoD efforts to acquire and sustain SiS capabilities and (2) the research-product environment in which technologies for building SiS products are conceived and developed. The objective of the SISPI is to influence the activities of the research-product environment to create technologies (methods and tools) that improve the cost-benefit-timeliness of acquiring and sustaining need-responsive SiS capabilities.

The purpose of the SISPI is to make routine development of SiS products predictable and to streamline predictable development. Its focus is on instituting effective means and methods by which acquirers and developers can predictably create products that meet current needs and evolve those products as needs change. The SISPI does not address technologies implemented by products or specific to the problems that developers must solve. The SISPI focuses narrowly on improving the methods and means by which products are created and evolved.

1.6.1 SISPI Summary Business Case

A business case answers four questions: what is to be achieved, why this is needed, how this will be accomplished, and whether its benefits are sufficient to justify its costs.

This characterization of the SISPI business case is limited in that it attempts only to argue that the state of producibility can be significantly improved through an advance of the sort described here. (A separate undistributed report provides an analysis of the economic arguments for investment in this vision.[1]) This discussion does not assume that there is a fixed date by which this vision must be attained to be worthwhile or that this effort to improve will ever be "finished." Rather, the case for SISPI is that opportunities for improving SiS development exist today, that improvements will be prolonged in attainment, and that new opportunities will arise over time as capabilities improve.

This roadmap as a whole describes what the SISPI is to achieve. It offers a unifying vision that is analogous to the transition from craft to manufacturing. Today's development of an SiS is a craft, highly dependent on the knowledge and skills of individual engineers. The SISPI vision of SiS development is manufacturing based on institutionalized knowledge and capabilities.

This effort is needed because it is widely agreed that the time, expense, and effectiveness of SiS acquisition and sustainment are problematic. A significant aspect of this issue involves a belief that the practices of systems and software engineering lack sufficient discipline and precision and that the products resulting fall short of reasonable expectations, with associated excessive delays, flaws, and rework.

The SISPI vision will be achieved through an integrated effort of research and active transition of technology into practical use. The roadmap for this effort is notional in the sense that technological advances cannot be mandated but depend on insight and enabling achievements by the scientific and engineering communities. As advances are attained, the roadmap will be recalibrated and revised to maintain its direction toward achieving the expressed vision.

The SISPI will provide benefits of substantial value to the DoD in that the systems and software engineering capabilities and challenges of SiS product development and evolution will become

[1] Turner, R., et al. SISPI Cost/Benefit Return on Investment Analysis

much better defined and understood. The awareness and consideration of opportunities for and implications of alternative solutions will become a routine facet of program planning and management decision making. The ability to evaluate an emerging solution much earlier and repeatedly as experience grows will lead to a better understanding and ability to manage unknowns, uncertainties, conflicts, and changes in needs. This will, in turn, reduce the costs and risks incurred by DoD suppliers and enable a more effective collaborative approach to SiS product development and evolution.

Just as advances in commercial technologies benefit the DoD, SISPI will result in technology advances that benefit commercial industry. Few if any of the envisioned technologies are uniquely applicable to DoD problems but will become viable for commercial industries as they mature. Some of the envisioned advances could come without the focus of an SISPI but the demands of DoD problems will exceed the capabilities that result if those problems are not considered in the conception of these technologies.

1.6.2 SISPI Critical Success Factors for Acquisition and Sustainment

The DoD enterprise of SiS product acquisition and sustainment is itself a large and complex system, operating under substantial burdens of cost-value accountability, complex evolving needs pushing the limits of technology, and complex communication and coordination challenges. The success of this enterprise depends on many factors, none of which are wholly resolvable through producibility improvements but all of which can be enhanced through SISPI actions. Several factors are significant considerations in SISPI thinking:

1. product alignment to operational needs (clear concise definition of required operational capabilities [problem], consistent with available technical capabilities and resources [solution])
2. product fit to operational context (proper dependency relationships with external systems/programs/organizations, formalized by definition and evolution of data/operational interfaces)
3. enterprise ability to anticipate and accommodate varying needs (changes, uncertainties, diversity, emerging opportunities, competitive challenges and risks)
4. alignment of performance to plan (transparency of progress through accurate tracking of performance to plan [objectivity, observability] and ability for timely replanning for adjustment of efforts [realism, responsibility, authority])
5. effective use of mature methods (integrated software and systems engineering methods and sufficient domain-specific problem/solution and technology expertise to resolve technical tradeoffs)
6. acquisition effort effectiveness (acquirer experience, domain knowledge, coherent charter, sound acquisition practices, programmatic interdependencies)

To enhance the success of the acquisition and sustainment enterprise, the SISPI must either orchestrate efforts that contribute to improving these factors or ensure that such efforts are realistic for the constraints associated with them.

1.6.3 Rationale for Producibility-Motivated Change

The following identifies perceived problems (or symptoms) in the development and evolution of SiS products. Most efforts using traditional development approaches exhibit one or more of these problems. For each described problem, an assortment of possible contributing causes is suggested. The goal of the roadmap is to target the most significant addressable causes with research and transition actions that lead organizations to use technologies (tools and techniques) that mitigate these problems. (Problems and causes are tagged with identifiers to allow for cross-reference in later justifying specific research/transition topics for consideration. The current list is presumed to be incomplete and in no particular order.)

<u>P1: Needs/Requirements—Statements of customer needs and product requirements at all levels fail to adequately define the problem or its context</u>

C1.1: Documents being too informal, developed in an undisciplined ad hoc fashion, by multiple authors/reviewers who lack a shared conception of needs and how to describe them

C1.2: Customers and systems engineers lacking sufficient understanding of how their choices affect software and then how the resulting software will affect the system

C1.3: Software engineers having insufficient knowledge and understanding of the customer's enterprise, mission, and needs

C1.4: Customers or systems engineers failing to expose uncertainties and potential for change (in needed capabilities or enabling technology), these being significant for software design tradeoffs

C1.5: Documents meant to communicate needed capabilities and constraints but whose content is ambiguous, overly detailed, incomplete and poorly organized

C1.6: Customers, acquirers, systems engineers, software engineers, and test engineers lacking shared concepts and vocabulary sufficient to communicate effectively

C1.7: Failing to consider and manage/accommodate ongoing changes in customer needs, technology, and understanding

C1.8: Requirements describing narrowly a solution (how the system should behave) rather than the problem (what capabilities are needed) and what criteria are to be used to judge acceptability of a yet-to-be-determined problem-solution

C1.9: Requirements defining software behavior under nominal (normal best-case) conditions but failing to define it for abnormal (constrained or failure) conditions

<u>P2: Testing—Testing, while consuming a large portion of development efforts, provides only a weak level of confidence in the quality or validity of a product</u>

C2.1: The limitations inherent to testing of being able to demonstrate only the presence, not the absence, of errors

C2.2: The impossibility of testing all possible uses of a product

C2.3: The nature of testing as a weak and time-consuming method for evaluating product quality

C2.4: Lack of intrinsic mechanisms for detecting and exposing the internal changing state underlying externally observable software behaviors over time

C2.5: The inability to determine through testing that a product will not enable unwanted functionalities

C2.6: The amount of effort required to set up a testing environment, create a suite of applicable tests, and repeatedly perform those tests and evaluate the results

P3: Management—Developers fail to adequately control for function, quality, and cost-schedule in creating a product

C3.1: Inaccurately estimating or failing to allocate expertise and effort needed to develop identified software capabilities

C3.2: Inability to precisely define or measure the non-functional qualities required of a product

C3.3: Failing to adjust plans to account for delays or effects of effort-function-quality tradeoffs

P4: Acquisition—Acquirers fail to properly establish and dependably manage to criteria for a fiscally sound, timely, mission-responsive acquisition-sustainment life cycle

C4.1: Ineffectiveness of secondary sources (expert commentary, documentation) as a medium for monitoring progress on a product

C4.2: Failing to give proper weight during acquisition to sustainment needs related to future changes to a product (e.g., potential for changes in mission needs or enabling technology or for operating cost or quality improvements)

C4.3: Overemphasis during acquisition on product capabilities over evolving customer/mission needs

C4.4: Lack of acquisition process orientation to revisiting prior decisions and replanning (redirecting effort, problem/solution) as circumstances change

C4.5: Lack of acquisition criteria and mechanisms for accommodating diversity in the problem/solution associated with a need for multiple product versions with differing capabilities

C4.6: Lack of acquisition responsibility for missions and systems above the level addressed by a problem/solution-defined product (or product family)

C4.7: Lack of acquisition policy guidance on when and how to adopt improved methods and tools, reinforcing a natural tendency to use only familiar methods and tools.

C4.8: Acquisition policy/practices that deincentivize suppliers from adopting productivity enhancements by rewarding direct effort in preference to capital investments in relevant problem/solution knowledge and capabilities

P5: Design—System level design does not properly position/frame or reflect software architecture factors and effects

C5.1: System engineers failing to adequately expose and weigh how system-level decisions affect software alternatives

C5.2: System design tradeoffs failing to reflect the effects of changing software decisions throughout the system life cycle

C5.3: Insufficient use of results from prior work as standard engineering practices (inadequate priority to using/adapting validated solutions to familiar problems rather than developing from scratch)

C5.4: Traditional engineering methods failing to provide effective techniques for building and maintaining systems whose requirements and technology change over time or that must be deployed in multiple versions

C5.5: Lack of systematic identification of alternative solutions and analyses of tradeoffs as a routine part of software engineering efforts, resulting in first-to-mind solutions that are revised in a trial-and-error fashion only after a problem is encountered

P6: Qualities—Characteristic and emergent properties of a system are not able to be determined prior to and as an influence over design but are achieved only through repeated trial-and-error improvements

C6.1: An overdependence on build-and-evaluate as an ad hoc means of achieving needed system qualities, reflecting an inability to build-in/predict and adjust qualities through systematic analysis and engineering of alternatives

C6.2: Inability to produce and comparatively evaluate alternative solutions to determine how properties/behaviors vary under them

C6.3: Lack of understanding and consideration of non-functional qualities as a primary consideration in requirements and design activities of software engineering

P7: Technology—Current development tools and methods impose practices, integrate poorly, and lack tailoring mechanisms to properly serve the needs of SiS products

C7.1: Slowness of introducing new or changing methods into tools

C7.2: Failure to design and implement tools in a way that allows tailoring to developer practices

C7.3: Lack of product-quality technology that supports emerging/unconventional development approaches

C7.4: Lack of active transition efforts within the DoD or suppliers for the introduction of improved development technology

C7.5: Lack of uniform criteria or trusted agents for establishing the utility and effectiveness of externally provided methods and tools, individually or in combination

P8: Legacy—Existing, deployed SiS capabilities are costly to modernize and risky to redevelop

C8.1: Difficulty moving legacy software onto a new platform

C8.2: Cost of reengineering legacy software to achieve improved properties

C8.3: Sensitivity of legacy implementations to platform/operational-environment behaviors

C8.4: Difficulty in being able to reverse engineer or fully understand decisions that led to a legacy solution

P9: Sustainment—Decisions made to expedite initial development of a product result in excessive costs and limitations on change as needs evolve

C9.1: Acquisition decisions limiting consideration of likely future needs in order to expedite initial delivery of capability

C9.2: Changes required that undermine design/implementation assumptions or architectural/interface integrity

C9.3: Lack of involvement of system life-cycle stakeholders in evaluating product development tradeoffs

1.6.4 Unrealized Advances

One influence on the SISPI approach is the perception that significant past research results have failed to transition successfully into large-scale practice. For these areas, the SISPI will seek to identify these, determine the causes, and promote efforts to improve and transition these results into greater use.

1.6.5 Related Efforts

The SISPI is only one of several related efforts that would benefit from cooperation and shared insights. These include other government-funded technology efforts (e.g., BAAs and SBIRs from DARPA, NSF, and NASA), ongoing university and defense industry research, non-U.S. R&D efforts [ITEA 2005, ARTEMIS 2005], other proposed efforts for technology advancement [Northrop 2006, Forrester 2006, Rajkumar 2007], and the many efforts of technology suppliers (commercial and open source). The intent of SISPI is to influence and leverage such efforts while avoiding unnecessary duplication so that the entire community of SiS researchers and developers can benefit effectively from any advances.

2 A Framework for SISPI Research

The purpose of the research framework is to identify areas of research that are needed to achieve the producibility reference vision, while delivering advances that provide near-term improvements in current practice. The value of the program will be judged based on its achieving both near-term improvements and progress toward the vision.

The producibility vision leads to five themes as a focus for research:

1. Model-based development (MbD)—Bridging the conceptual gap between domain experts and product developers
2. Predictable software attributes (PSA)—Building software and systems whose properties are predictable and adjustable
3. System virtualization (SV)—Enabling pre-verification of the real-world behavior of software and systems
4. Disciplined methods (DM)—Achieving engineering discipline in the interdependent development of software and systems
5. Infrastructure and emerging technology (IET)—Adapting producibility advances to exploit or accommodate changes in infrastructure and enabling technologies

Each research or transition theme is described by a set of objectives, encompassed topics, and notional milestones, with explanatory elaboration. This constitutes a framework within which researchers, tool developers, and transition agents may propose relevant efforts. Milestones listed under each theme are meant to be only suggestive of needed research or transition efforts and are in no particular order at this time. The sets of milestones all require significant further development and elaboration. The intent of the form of these items is to begin to describe a network of interdependent topics. "PYxx" indicates a targeted program year to suggest possible timing priorities for a milestone. Each milestone is identified with a prefix code (R-XXX-t-n), in which XXX identifies the theme, t when present identifies a topic, and n is a sequence number. Tags in trailing square brackets point to items that are thought to be prerequisite to that item.

2.1 MODEL-BASED DEVELOPMENT

The Research theme of Model-based Development (R-MbD) is concerned with representing a problem and associated solution in the form of a precisely defined modeling notation. A properly conceived problem-solution model provides the information required to derive a needs-responsive SiS product. An effective modeling notation provides the means to specify alternate problem formulations and, for each of these, a set of alternative solutions. A model-based development (MbD) capability provides the means to identify and evaluate the effects and interactions of alternate problem-solution formulations of key customer and engineering decisions to create a conforming product. A modeling notation must have unambiguous semantics to a domain expert and be sufficiently complete to allow mechanical derivation of a product; any ambiguity is in fact resolved by the product derivations that result. As customer needs and technology evolve, an MbD capability supports revising an existing problem-solution model to rapidly produce a revised product.

2.1.1 Definitions

model	a representation (of a product) that enables approximate answers to a designated set of questions (about the product)

2.1.2 Objectives

- Bridge conceptual gaps, including differing terminology and assumptions, among acquirers, domain experts, systems engineers, and software engineers.
- Provide a unified expression of all facets of a problem as a perception of needed capability and its realization as alternative potential solutions.
- Condense the dialogue for converging on shared understanding and expression of a responsive problem-solution, focused on key needs-driven decisions.
- Enable rapid exploration of alternative solutions as a means to improve understanding of the problem and the factors that determine solution fit.
- Accommodate continual change in customer needs, enabling technology, or understanding.
- Enable rapid generation of a product over its life cycle from an evolving problem-solution model.

2.1.3 Topics

1. Representation (R)
    a. What information is required to adequately represent a problem-solution model?
    b. What information is needed to represent the model-product relationship?
    c. What information is required to support analyses of problem-solution and product properties?
2. Problem analysis and specification (P)
    a. What forms of expression are effective as means for domain experts to characterize a problem?
    b. What mechanisms are needed to enable collaboration among multiple domain experts in describing a problem?
    c. How are uncertainties and potential changes accommodated in problem descriptions?
    d. If there are multiple viewpoints of a problem, how are these kept consistent?
3. Solution analysis and validation (S)
    a. What capabilities are needed to gain insight into how a problem description determines a solution?
    b. How are the implications of alternate descriptions of a problem represented for analysis?
    c. What capabilities are needed in support of validating that a modeled problem-solution will in fact satisfy customer needs?

4. Product Generation (G)
    a. What information must be added, beyond that used to describe the problem and solution, to generate a product?
    b. What mechanisms are needed to generate all facets of a product given information represented in a problem-solution model?
    c. How will a product generation capability be validated as properly maintaining model assumptions without unnecessarily limiting the resulting design-implementation?
5. Model-product verification (V)
    a. Given a model and a corresponding product, how do we verify that the behavior and properties predicted by the model are in fact consistent with the product's actual behavior and properties?
    b. Given a model-product discrepancy, how do we locate the source of the discrepancy in the model or the model-to-product transformation?

2.1.4 Notional Milestones

(PY2) R-MbD-1	a capability to express a problem in a notation and terminology that a specified community of domain experts can use effectively to describe, discuss, and resolve alternatives, uncertainties, and tradeoffs [R-DM-R-1]
(PY4) R-MbD-2	a capability that can be specialized for and used by domain experts to satisfactorily specify a problem in their domain, sufficiently that experienced developers can build a conforming product without further information [R-MbD-1]
(PY5) R-MbD-3	an integrated model-based development capability, appropriate for general use or for use through customization in multiple domains []
(PY3) R-MbD-4	a data management capability sufficient to enable retention and retrieval of model-based specifications in forms suitable for activities of model-based development [R-MbD-8]
(PY3) R-MbD-5	a capability to mechanically construct a customized product that conforms to the meaning implied in the use of a chosen problem-solution modeling notation []
(PY5) R-MbD-6	a capability that domain experts can use to specify the essential aspects of a problem and obtain a customized product that provides a solution [R-MbD-2, R-MbD-5]
(PY6) R-MbD-7	a capability that domain experts can use to obtain and compare multiple customized products as alternative solutions that satisfy the essential aspects of a specified problem []

(PY1) R-MbD-8	a conceptual taxonomy and schema for expressing the content of a general or domain-specific modeling notation []
(PY1) R-MbD-9a	definition of a comprehensive notional method for tool-neutral performance of model-based development []
(PY2) R-MbD-9b	a demonstration construction, using open-source or commonly available tools, of a minimally complete model-based development capability for a narrow but broadly DoD-applicable domain [R-MbD-8, R-MbD-9a]
(PY4) R-MbD-9c	a productized construction of an optimally complete model-based development capability in each of three DoD-applicable domains [R-MbD-9b]
() R-MbD-10a	extensions of demonstration constructions to integrate PSA advances as problem or solution analysis elements of model-based development []
() R-MbD-10b	extensions of demonstration constructions to integrate SV advances for more comprehensive solution validation facilities and more realistic results []
() R-MbD-10c	extensions of demonstration constructions to integrate DM advances for more effective system/software engineering methods and associated technologies []
(PY5) R-MbD-11a	a productized open-source framework and tool components for constructing model-based development capabilities [R-MbD-9c]
() R-MbD-11b	extensions of open-source framework and tool components to reflect improvements due to PSA and SV advances [R-MbD-11a, R-MbD-10]

2.1.5 Elaboration

Model-based development is the framework and constructive mechanism for the manufacturing of SiS products. It provides the means by which alternative problem-solutions are described in model form, evaluated, and transformed into a product.

The primary focus of R-MbD is the provision of a multi-faceted, multi-level model that enables communication among customers who have a need and engineers who have the expertise to create a product. Minimal levels of MbD capability are feasible with existing technology. R-MbD efforts focus on increasing the leverage provided by such MbD mechanisms. Technologies resulting from other producibility research themes enhance MbD capabilities, particularly in terms of understanding the derived properties of a product and its behavior in an environment.

Formulating a Product

Conceptually, R-MbD envisions the formalization of a product-defining problem-solution and its context of use as *projections* of domain knowledge into three semantically linked expressions:

1. the capabilities, in domain-specific terms, that a customer wants to gain with the product and any environmental (intrinsic) or enterprise (extrinsic) constraints on how that product is to be constructed or used
2. formulations of the problem and associated solutions that approximate the needed capabilities
3. extrapolations of the problem-solution formulations in terms of the behavior and properties that the product will exhibit in use and its effects on the system and ecosystem of use

The characterization of the product and context in each of these projections can be independently changed as long as consistency among them has been established and is retained or subsequently repaired.

Projection 1 formalizes customer conceptions of the needed product as a set of choices that embody an enterprise's mission, strategy, business approach, and organizational capabilities and resources. It accommodates expressing flexibility in terms of uncertainties, alternatives, and unknowns. This projection drives conception of the other projections and, as it changes over time, it drives changes in the other projections; it in turn changes in reaction to insights gained from work on the other projections. Intrinsic constraints express the nature of the environment in which the product is to be used. Extrinsic constraints indicate enterprise policies or choices that are outside the product acquirer's scope of control.

Projection 2 corresponds to traditional conceptions of product construction, encompassing formulation of requirements, design, implementation, and verification based on customer needs, as expressed in projection 1. The conception of projection 2 is that there are multiple problem formulations that could to varying degrees satisfy projection 1 and multiple solutions that would satisfy each problem formulation. Each resulting problem-solution formulation of the product is an approximation to the customer formulation of needs. In a product line context, this projection provides for derivation of previously developed product formulations customized to projection 1 content.

Projection 3 provides the means to choose among alternative problem-solution formulations of projection 2 to find the best fit to needs and constraints expressed in projection 1. Such choice depends on exposing and evaluating implications and tradeoffs in how each problem-solution formulation affects the operational context of the targeted system and ecosystem of operation. This requires representing the behaviors and properties of the system and ecosystem contexts that affect and are effected by the product and being able to monitor and evaluate the dynamic effects between the product and its context.

Descriptive Modeling

R-MbD envisions the use of domain-specific problem-solution models that facilitate focused communication between customers and developers for the efficient construction and timely evolution of effective products. The traditional prescriptive approach of formulating a problem in text

and a solution in a programming language is a weak medium for communicating between developers and customers, resulting in poorly understood problems and ill-fitting solutions.

A descriptive model enables expressing a problem and constraints on a solution from a customer perspective in a form from which developers can systematically derive and evaluate alternative solutions. One implication of such a model is that, at the model level, the distinction between software and systems, engineering is blurred. The model structure itself reflects the structure of how customers perceive the system but software is a pervasive element in giving that structure and its elements meaning. This provides the basis for representing a product (and its environment) entirely in software, allowing hardware choices to be made for purposes of behavioral optimization rather than by default or to be deferred until required specialized hardware is available.

Product structure and transformations from model to product are predetermined in the conception of the model notation itself. Many of the elements of a complete product are implicit in a descriptive model, either assumed by convention or included dependent on other customer-level choices. Advanced product line approaches, limited in application to families of similar products (http://www.domain-specific.com), have envisioned such a capability, analogous to the manufacturing concept of mass customization. R-MbD is conceived as an effort to generalize such approaches to the conception, engineering, manufacture, and evolution of any product.

A descriptive model is necessarily incomplete in that its purpose is to allow derivation of many alternative problem-solution formulations. A customer's needs are often poorly understood and changing over time. A descriptive model permits the expression of such uncertainties and instabilities that traditionally are fixed arbitrarily early in development to avoid the difficulties of having an "incomplete" statement of requirements. With descriptive modeling, incompleteness and alternatives are important mechanisms for engineering flexibility needed to create multiple potential solutions that can be systematically and empirically compared and refined for best fit to customer needs.

A preferred problem-solution formulation is determined through a process of developing and evaluating alternatives that resolve uncertainties and tradeoffs in different ways. Alternative formulations can be produced to help customers resolve uncertainties and fix unstable factors. When multiple formulations satisfy a customer's needs, further tradeoffs can be made based upon secondary costs and benefits to determine a preferred product formulation. By reducing incompleteness and eliminating alternatives, the number of formulations is reduced; similarly by retracting decisions previously made, the number of formulations is increased, resulting in a greater diversity of product being possible.

Each problem-solution formulation is an approximation of the product needed; the formulation that is judged to be the best approximation in the time available is applied to build the corresponding product for customer use. That formulation then serves as the basis for future product versions as changes are identified in it that provide a better approximation of the customer's changing needs.

2.1.6 References

2.2 PREDICTABLE SOFTWARE ATTRIBUTES

The Research theme of Predictable Software Attributes (R-PSA) is concerned with measuring, predicting, and controlling significant properties of an SiS product. Principal focus is directed to properties that are determined or affected by decisions about software as they arise in addressing customer needs (specifying a problem-solution) or applying engineering judgment (deriving a product that satisfies a problem-solution).

2.2.1 Definitions

design	formulation and analysis of problem-solution alternatives, weighing uncertainties and tradeoffs that determine the effectiveness of the resulting product
tradeoff	an interaction among the mechanisms that determine two or more properties of a product's behavior
uncertainty	a probability that unknowns or extrinsic circumstances (opportunities and risks) will cause the effects of a product's behavior on a system to diverge from its expectation

2.2.2 Objectives

- Establish a comprehensive and consistent reference taxonomy of system and software properties that affect the acceptability of a product.
- Provide the means to measure and model all significant properties of a software product and interactions among them.
- Provide means to predict measures of product properties based on the characteristics of alternative solutions.
- Provide means for transforming solutions to modify associated properties with and without affecting functionality.

2.2.3 Topics

1. Identification (I)
    a. What are the critical properties of interest for an SiS? (e.g., performance, reliability, availability, security, safety, usability, ...)
    b. How precisely are each of these properties defined and what factors indicate the importance of each in a given SiS?
    c. What are the interdependencies among properties and how do direct changes in a given property implicitly affect others?
2. Analysis (A)
    a. How is each of the relevant properties of an SiS to be expressed?
    b. What measures can be used to indicate the condition or changing state of each property?

   c. What are the tradeoffs (e.g., effort, effects on product behavior or other properties) in choosing how to measure, or not to, each property?

   d. What means are available (e.g., visualization) to portray the aggregate state of a property that exhibits as localized effects in a product?

3. Prediction (P)

   a. What are the means by which software choices affect SiS properties?

   b. What is the correlation between particular software choices and the measures that characterize each property?

   c. How can changes to improve one property be understood in terms of its implications for other properties?

4. Optimization (O)

   a. What means are available to adjust requirements, design, or implementation so that properties are affected in a predictable way?

   b. How do property targets affect developer choices?

   c. What means are available to inform developers on implications for properties of changing a requirements or engineering choice?

   d. How can tradeoffs among properties be visualized, supporting achieving the best combination of properties for the product?

2.2.4 Notional Milestones

(PY1) R-PSA-1	a comprehensive reference taxonomy of the potential properties of interest needed to fully characterize the nature of a system []
(PY2) R-PSA-2	a systematic evaluation of the state of theory and practice in the ability to measure, predict, and control each property of a system [R-PSA-1]
(PY3-4) R-PSA-3	a systematic analysis and formulation of the interdependencies among the properties of a system [R-PSA-2]
(PY3-8) R-PSA-4	targeted advances in the ability to measure significant system properties [R-PSA-2]
(PY3-8) R-PSA-5	targeted advances in the ability to predict measurable system properties [R-PSA-4, R-PSA-3]
(PY3-8) R-PSA-6	targeted advances in the ability to control predictable system properties [R-PSA-5, R-PSA-7]
(PY5-10) R-PSA-7	targeted advances in determining and controlling tradeoffs among interdependent system properties [R-PSA-3]

(PY1-3) R-PSA-8	high-value improvements in the near-term utility of existing tools that support measuring or predicting significant system properties [R-PSA-1]
() R-PSA-9	mechanisms for depicting simplified views of the properties of a product in aggregated and filtered forms according to priorities in customers' needs []

2.2.5 Elaboration

Many important properties of a product are indirect derivatives of its construction or its interactions within a system and environment. Today, we only poorly understand how to predict, and in some cases even measure, many of these properties. Typically, we can determine that a property is unacceptable only by observing the operational behavior of a system in use. Complicating our ability to control a product's properties is that actions taken to modify one property can change the limits on other properties. Therefore, the determinants of not only individual properties but also the interdependencies and tradeoffs among properties must be understood and weighed. The purpose of the PSA theme is to improve engineers' ability to identify, measure, predict, and mutually adjust the properties of an SiS product earlier, more easily, and more accurately during its development.

A major challenge in beginning to improve current practice is to establish a systematic treatment of important system and software properties as part of requirements and design activities. The effects of systems engineering decisions on software complexity and the effects of software on system properties are often recognized and addressed, at significant cost and disruption, only after an implementation has been produced. Within the limited context of DoD SiS, it should be possible to establish a reference taxonomy of properties to be considered and to develop a system-independent understanding of how tradeoffs among these properties affect product capabilities. Acquisition programs should be required to identify the significance of each property for the planned product and how those properties and their interdependencies are affecting the design of the product.

Recognizing that many software properties are inadequately quantified and predicted and that the implications on properties of design and implementation choices are not well understood, significant effort is envisioned on how to formalize properties and the factors and decisions that affect them during a development effort.

Under an MbD approach, the product model is descriptive, defining the conception that a customer has of needed capabilities. For PSA, there is the need for derivative models that give insight into the effects of product decisions on SiS properties. Each property of interest for an SiS may be characterized by multiple models, each focused on different facets or perspectives of a property or interdependent set of properties. Each model should be configured in terms of the factors representing customer needs and problem-solution decisions that derive from customer and engineering judgment.

2.2.6 References

Barbacci, Mario; Klein, Mark; Longstaff, Thomas; & Weinstock, Charles. *Quality Attributes* (CMU/SEI-95-TR-021). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1995. http://www.sei.cmu.edu/publications/documents/95.reports/95.tr.021.html

2.3 SYSTEM VIRTUALIZATION

The Research theme of System Virtualization (R-SV) is concerned with enabling the pre-verification of products in a virtualized SiS environment. In particular, to pre-verify a product, the environment must contain the other elements of the targeted system in operational (when feasible) or simulated forms. To the degree that product behavior involves sensing or effecting elements of the real operational environment, the environment and its elements may need to be emulated/simulated, either because these elements cannot be physically realized or to enable inducement of sub- or super-realistic behaviors. A key implication of virtualized pre-verification is the need for the ability to observe and control the internal behavior (e.g., rate of operation, internal information state) of the product at all levels and of the operational environment. This requires the ability to instrument the product and simulated environment elements without incurring indeterminate effects on the behaviors of each.

2.3.1 Definitions

operational environment	the observable information space that an SiS detects or affects as it operates.
platform	the computing environment (hardware and software) into which an SiS product is installed and operates

2.3.2 Objectives

- Provide the means to construct a virtual environment that adequately models the behaviors of a potential operational environment.
- Provide the means to emulate hardware devices and to dynamically modify the characteristics and state of such devices.
- Provide an experimental capability in which time is an independently controllable variable.
- Provide the means to inspect and modify the operational state of software operating within a virtual environment.
- Provide the means to model the behavior of interacting systems or people through recording or condition-based scripting that permits automatic repetition of the resulting interactions with software being evaluated.
- Provide the means to selectively capture the dynamic state of the simulated system as it operates, for subsequent analyses.
- Provide the means to concurrently operate and compare states and outputs of multiple alternative implementations of a modeled system.

2.3.3 Topics

1. Platform independence (P)
2. Hardware abstraction (H)

3. Environment simulation (E)
4. Usage simulation (U)
5. System validation (V)
6. Integration (syntax, semantics, pragmatics) of communicating tools (I)

2.3.4 Notional Milestones

(PY2) R-SV-1	design for a family of virtualized system validation facilities [R-SV-2]
(PY1) R-SV-2	prototype validation facility installations, each for a family or narrow class of SiS or subsystem, within which software can operate on a targeted platform with simulated hardware devices []
() R-SV-3	shared sources for previously developed hardware device simulations []
() R-SV-4	capabilities for creating hardware device simulations []
() R-SV-5	capabilities for creating a simulation of an operational environment []
() R-SV-6	capabilities for simulating the (normal and degraded) operations of a collection of SiS users, devices, and external systems []
() R-SV-7	Mechanisms for exposing non-observable software state and operation []
() R-SV-8	capabilities to transparently exchange real and simulated hardware devices within a validation facility []
() R-SV-9	capabilities to generate hardware device fabrication specifications from device simulation specifications []

2.3.5 Elaboration

The objective of acquisition is to provide products that improve an enterprise's ability to perform its mission. This ability is a function of the capabilities of the product being acquired, the capabilities of other system elements (people, hardware, and systems as determined by other products), and the behavior of other systems that comprise the mission ecosystem. Perfect understanding of whether and how all of these behave in aggregate under all circumstances is impossible and not fixed over time. The value of SV is to provide technologies that help the acquirer and developer understand how to create a product that comes closest with current understanding to giving the customer enterprise the capability that it needs.

SV envisions a framework for constructing a virtual environment into which a product can be injected just as it would be into an actual operating environment and evaluated against expectations. A virtual environment is inferior to the real environment in that, as with any model, it abstracts details that can lead it to give inaccurate results under particular conditions; however, a virtual environment is superior for purposes of validation because it is a controlled environment in which effects of product behavior can be contained within that environment and controlling, monitoring, and measuring of the product's behavior and effects can be more penetrating and pervasive when needed.

This virtual environment should be a hybrid composition of actual environment elements to the degree feasible and emulated or simulated versions of other elements as needed to allow the product to be used as envisioned. Virtual environment elements may be emulated/simulated or encapsulated as needed to allow the imposition of instrumentation for monitoring of the otherwise unobservable (internal) behavior of those elements or to provide extended control over those behaviors. The virtual environment itself must provide control over time as system elements detect it so that behaviors can be delayed or expedited.

As SV capabilities advance, the MbD framework must be adapted as needed to accommodate integrated use of these capabilities.

2.3.6 References

2.4 DISCIPLINED METHODS

The Research theme of Disciplined Methods (R-DM) is concerned with achieving engineering discipline in the construction of SiS products. Systems and software engineering practices are only as reliable as the methods underlying them. Today, those methods are largely focused on recording the results of engineering analysis and decision making, automating what are primarily clerical tasks and doing little to enhance substantive engineering tasks. Better practice requires more effective methods that reflect both the substantive elements of engineering and the practical limitations of real-world constraints of time, cost, and complexity.

2.4.1 Definitions

method	guidance and criteria that prescribe a systematic, repeatable technique for performing an activity
methodology	an integrated body of principles, practices, and methods that prescribe the nature and proper performance of a process
process	a partially ordered set of activities or actions, conceived as a means of accomplishing specified objectives

2.4.2 Objectives

- Create technology that enables attainment of the SiS vision as elaborated through topics of other research themes.
- Create methods and tools that integrate within a methodology for a coherent and cohesive process of product engineering and manufacture.
- Elaborate methods and tools that accommodate and reduce uncertainty and ambiguity in a developing product.
- Create methods and tools that accommodate early practical use of capabilities enabled through other research themes.

2.4.3 Topics

1. Management (M) *{planning, monitoring, assets, risk, reporting}*
2. Process (P) *{methodology, documentation, measurement, quality, automation}*
3. Requirements (R) *{needs, scenarios/uses, variability}*

4. Design (D) *{architectural: decomposition, concurrency, dependency; component: interfaces, internals}*
5. Implementation (I) *{languages, adaptability, reverse engineering}*
6. Verification and validation (V) *{testing, review, formal analyses}*
7. Systems engineering (S) *{hardware-software co-design, system-software codependency}*

2.4.4 Notional Milestones

(PY2) R-DM-M-1	a DoD-adoptable standard for software metrics by which acquisitions are to be managed, and uniform guidance on tolerances for when out-of-bounds measures must trigger risk mitigation actions, such as replanning []
(PY1) R-DM-M-2	guidance on commonly experienced software risks that are to be actively managed in all DoD Acquisition programs [R-DM-M-1]
(PY1) R-DM-P-1	a repeatable method for transparent monitoring of an iterative software process and progress reporting within the framework of a linear phased acquisition/system engineering effort []
(PY1) R-DM-R-1	a practical (generic or domain-specific) representation and method, usable by domain experts and/or systems engineers, for the expression of system and software requirements []
(PY2) R-DM-R-2	a method that facilitates systems engineers in identifying and characterizing the nature of all system-level constraints that affect software [R-PSA-1]
() R-DM-I-1	a method and notation for representing implications of component family variabilities as a factor in deriving formal proofs of instance component properties []
() R-DM-V-1	techniques for estimating product properties based on architectural representations []
() R-DM-V-2	technology for rapidly creating a customized testing environment based on standardized infrastructure capabilities and problem-specific operational scenarios []
() R-DM-V-3	an effective method for properly peer-reviewing the conformance of development work products to specifications []
() R-DM-S-1	means to represent any hardware device in a computational model sufficient to enable simulated use, prior to fabrication, as a functioning element of a system in conjunction with other (physical or virtual) system elements []

() R-DM-S-2	techniques for representing computations in a form that can be interpreted as a specification for either hardware fabrication or software generation (or symbolic execution) []

2.4.5 Elaboration

Methods are the foundation upon which a methodology is built. Conventionally, methods guide how developers accomplish the tasks that create the constituent work products of a product. Following convention, DM topics are organized into a familiar set of method categories. Within the DM context, the premise is that methods can be conceived and improved somewhat independently of how they fit and are used in a particular process or methodology, with the likelihood that methods will require tailoring for use in a specific process or methodology (such as envisioned in the 'process lines' concept of the IPRC process research framework [http://www.sei.cmu.edu/iprc] or the process adoption method of a product line methodology [http://www.domain-specific.com]).

A model-based methodology incorporates methods as the mechanisms that give meaning to a model. The DM theme starts from the premise that an understanding of performing a task manually precedes an ability to automate or mechanize that task. DM topics that enhance manual performance of tasks have the advantage of providing near-term value in conventional development efforts but must at the same time provide insight into mechanisms that an MbD capability requires.

The purpose of management is to ensure that resources are applied effectively so that an acceptable product is deployed into use within a reasonable timeframe. DM focuses on improving practices of planning, of identifying and adjusting to uncertainties (opportunities and risks), of monitoring progress that leads to replanning, and of reporting on progress to higher management. Supporting responsibilities include maintaining consistent versions of all project and product artifacts.

Process first concerns how an enterprise works, in the case of DM the result being an effective properly tailored methodology for the purpose of creating a product. Secondly, Process concerns secondary activities that provide visibility into the other essential activities of product engineering and manufacture. The DM concern for this is both to improve the transparency achieved and to reduce the effort involved in achieving it. The activities of interest include documenting the results of project activities as work products, measuring the effort of performing activities for purposes of improving them, and evaluating the quality of activity results for ensuring adherence to prescribed practices and for insights into better practices. Insights on automating aspects of activities can arise from such process efforts.

Requirements for DoD SiS products are typically expressed in the form of "shall" statements. These are typically not well organized nor verifiably complete. Often these statements go beyond a statement of needed capabilities to describe a particular solution or approach based on past experience. Further these are often required to be "final" prior to development rather than being refined and evolved as issues are discovered during development efforts. This flawed approach to requirements has arisen because requirements are used as a primary means of legally defining the bounds of a contracted effort. DM envisions that the current conception of requirements will be reformulated as "needs" which are a minimal statement of needed capabilities and constraints envisioned by the customer and "requirements" which are an evolving specification of the verifiable

properties of a product that must be met for it to be delivered into operational use as part of a system.

Design is the conception and structuring of a solution that satisfies the criteria expressed in specified requirements. The challenge of design is not to envision a single fixed design but rather to conceive a design that will accommodate likely changes in requirements and engineering tradeoffs over time. Design is a concern at two levels: architecture and component. Architectural design is concerned with the decomposition of a system logically (into components), structurally (functional and information dependencies among components), and temporally (timing and concurrency). Component design is concerned with determining the information and functional content, internal structure, and interfaces of each component so that architecture-level decisions are achieved.

Implementation is currently conceived as constructing a single solution that satisfies fixed (but indeterminately changeable) requirements. A design is often subverted for implementation expediency, particularly during periods of "debugging" that usually occur under conditions of substantial stress. Implications for future revision of an implementation, including documentation of complex logic and engineering decisions and tradeoffs, are seldom given much thought. Modifications to such code often has unexpected side effects, sometimes on other indirectly related code. The focus under DM is to reorient implementation toward creating and evaluating alternative solutions that can then be selected for use as tradeoffs warrant. DM envisions flexible multi-paradigm languages with integrated mechanisms for adaptability to differing needs and constraints without direct modification of essential logic. Support for renewal of legacy software requires better mechanisms for extracting its abstract implementation from which to derive better equivalent implementations, potentially in a different language.

Verification entails a combination of testing, reviews, and formal methods. The motivation in DM is both to improve the confidence in a product and to reduce the effort required to do so. Testing is known to be a very weak method of determining whether an implementation is free of defects and yet remains the primary means of doing so today. Tests are typically based on extensive anecdotal cases of how a product is used within a system. Given the impossibility of exhaustive testing, testing efforts take on an indeterminate, subjectively determined duration. Reviews of code, if performed at all, are often overly formalized and too focused on form over content and probing of decisions and implications for potential changes.

At the intersection of implementation and verification, metaprogramming methods provide for representing families of components in forms from which instance component implementations, associated formal proofs of properties, and customized documentation can be generated mechanically based on problem-level decisions. By working at the level of a component family, the effort to create consistent work products can be leveraged across all future instances of the family. In particular, the costs of applying formal proof methods and of modifying components to meet changed needs are more acceptable when applied in the context of a family.

Validation, establishing that a product satisfies a customer's actual needs rather than simply what has been requested, is distinguished from verification in that it focuses primarily on the uses of a product in its operational context. Under MbD, validation operates with reference to the product model and occurs throughout product development. SV efforts are a principle source of mecha-

nisms for more systematic validation of system properties in an MbD framework as well as for use in conventional development.

Systems engineering concerns focus on two primary issues: (1) flexibility in techniques for determining whether capabilities are realized in hardware or software, including the ability to defer or change such decisions or support multiple implementations, (2) better understanding and visibility concerning system-level decisions that affect software and software-level decisions that affect system properties. The first also has particular utility related to the SV research theme while the second will be informed by work in the PSA research theme.

2.4.6 References

2.5 INFRASTRUCTURE AND EMERGING TECHNOLOGY

The Research theme of Infrastructure and Emerging Technology (R-IET) is concerned with identifying SISPI-independent advances in infrastructure and enabling technologies and adapting producibility capabilities to exploit or accommodate those advances. Primarily, this theme focuses on identifying and determining the implications of evolving commercial technology as it affects producibility, leading to actions for the tailoring of the commercial or SISPI technologies to enhance producibility results.

2.5.1 Definitions

2.5.2 Objectives

- Identify advances in computer and communications technology that enable advances in producibility technology.
- Identify advances in computer and communications technology that change assumptions upon which producibility technologies depend.
- Create infrastructure technologies that support accommodation and exploitation by producibility technologies of other technology advances.

2.5.3 Topics

1. computational technology
2. software componentization, customization, and commoditization
3. communication, collaboration, and human interface technologies
4. complex data management

2.5.4 Notional Milestones

() R-IET-1	advances in exploiting concurrent (multi-thread, multi-core, multi-processor, networked, grid), virtualized, or mobile/distributed processing mechanisms []
() R-IET-2	advances in exploiting cooperative and collaborative communications mechanisms []
() R-IET-3	advances in operating under latency, with distributed data sources, asynchronous services, and autonomous agents []
() R-IET-4	advances in evolving continuously operating and adaptive software []

() R-IET-5 advances in persistent data management (representation, retention/recovery, schema evolution) []

() R-IET-6 advances in computing with dynamic (uncertain/fuzzy-valued, time-sensitive, streaming) data []

2.5.5 Elaboration

A premise of SISPI is that the acquisition and development of products can be improved within the framework of existing technology. However as this technology evolves, there will be new opportunities or constraining dependencies that arise.

The IET theme exists in recognition of the constant and continuing evolution of technologies comprising SiS and development infrastructures. As these technologies advance, development capabilities must exploit and accommodate them in order to best satisfy customer needs. Other producibility research must balance the need to provide value in the near-term against the need to work effectively in the future and the increased capabilities that those advances could enable. The purpose of IET research is to understand and evaluate the applicability of future infrastructure and emerging technologies as enablers or determinants of producibility technologies.

The emergence of multi-core processors and multi-processor systems are of particular concern from a software engineering perspective. Most development today occurs under an implicit assumption that software will execute under control of a single processor. Specialized applications exist that exploit multiple processing units (e.g., parallel processing of homogenous data streams) but in general software engineering does not provide adequate notations or tools for the construction of concurrently executing software. Advances that give developers the concepts and tools to build concurrently executing software may motivate significant changes in development methods and infrastructure.

The increasingly distributed nature of computing undermines simplifying assumptions that underlie most software today in terms of computational latency and data currency. As distributed capabilities become more prevalent, technology must enable developers to understand and account for delays and time lapses traceable to distributed data dependencies.

A related implication of increasingly distributed computing resources will be the need for better ways of working collaboratively on products across distances. Tools exist today for flexibly sharing access to work products remotely but methods that effectively exploit this capability as part of an integrated approach have not been adequately developed.

The traditional paradigm of software engineering was a dichotomy between high-cost custom products and highly replicated generalized products. With the conception of product lines, there is the potential for creating moderately replicated customized products. As the capabilities of technology supporting customization advance, the mechanisms used to build software will come to resemble manufacturing, requiring greater investment in infrastructure but leading to substantial reductions in per-unit production costs. This will affect not only the means by which similar products are tailored to differing needs but also the means by which each product is evolved as the needs of its users change over time.

A trend toward componentization of software offers the potential for lower cost customized products. Componentization is similar to the use of off-the-shelf products in that user needs and business process must be adapted to accommodate those products' capabilities, mechanisms, and design choices. By making such accommodations, the cost of producing and supporting a customized product is reduced, in turn tying that product to the future evolution of the commercial product even when that product changes in undesired ways or fails to add needed new capabilities. Componentization follows this approach at a more detailed level, providing capabilities at reduced cost but imposing design decisions and limiting tradeoff options. The potential increases for individual components to diverge from needs and preferences however open-source components may allow some degree of customization leading to higher maintenance cost. If customization of components is not controlled with a clear focus on architectural cohesion, a proliferation of similar components can result in difficulty identifying the best match for use in future products.

2.5.6 References

3 A Framework for SISPI Transition

"In theory, theory and practice are the same; in practice, they are not."
Jan L.A. van de Snepscheut

The purpose of the transition framework is to identify actions that will facilitate the transition of research advances into active practice. The value of the SISPI will be judged based on its achieving both near-term improvements in DoD/industry practice and progress toward attainment of the producibility vision.

Transitioning producibility technologies involves four types of activity as described earlier in the normative technology advancement life cycle:

1. validation—determining and demonstrating the applicability, adoptability, and practical value of research results for building DoD systems
2. integration—adjusting the scope, interfaces, data representations, and conventions of related methods and tools so that they produce consistent results when used in appropriate combinations
3. productization—engineering research results into integrated engineering tools and methods suitable for production use
4. adoption—facilitating the selection, introduction, and institutionalization of productized tools and methods by DoD acquisition programs

Technologies resulting from research will have different realizations, typically as methods, as tools, or as tool components. Validation, integration, and productization will proceed at different rates, depending on how a technology is realized and its degree of interdependence with other technology realizations. An SISPI technology transition authority will be constituted to monitor and guide the maturation of technologies from research results into productized methods and tools adopted by DoD acquisition programs and their industry systems/software engineering suppliers.

Validation involves researchers from universities and industry, working within the context of representative DoD problems, to evaluate the effectiveness and maturity of submitted research results. A research topic will have been defined in terms of the value it is intended to provide. Resulting research will typically produce incompletely defined technology that may require careful set up or problem constraints to work properly. Validation attempts to determine whether, within these limitations, the technology does provide the intended value; it further identifies any essential or desirable improvements that are prerequisite to efforts to enhance, integrate, and productize the technology.

Integration is a specific form of improvement that technology resulting from research will often require. Having been focused on achieving a particular advance, researchers may fail to provide functionality or interfaces required for integrated use within a complete production process or may include functionality that is provided separately in a production environment. These simplifying assumptions, once the technology has shown value, must be rethought to allow the technology to fit better within a production environment. This is complicated by the potential that a par-

ticular technology may need to be used in multiple contexts that differ in terms of how the technology is to fit; part of the validation and integration dialog will be a consideration of whether flexibility is needed or whether constraints are to be assumed in the applicability of the technology.

Productization involves researchers and product-quality tool developers working together to transform a technology into an appropriate production-quality method or tool. Tool developers may be a commercial vendor intending to offer the technology as part of a software tools business or a defense software supplier intending to adopt the resulting tool for use on appropriate acquisition programs.

Adoption is the interaction between providers of productized technologies and potential customers for those technologies. Principal customers are DoD acquisition and sustainment programs; secondary customers are the defense systems/software industry that provides engineering and technical support to these programs. It is a responsibility of each acquisition program, as part of the Technology Development phase of the DoD Acquisition life cycle, to identify and develop or acquire technology that will enable achievement of a cost-effective solution. Programs have historically neglected to give sufficient attention to needs and opportunities for properly provisioning system/software engineering activities in the way that hardware components are, as manufactured goods.

Adopting producibility technologies will require an effort by each acquisition program to identify their needs for effective system/software engineering and the systematic introduction of technologies that address those needs. This will be accomplished by the commitment of acquisition program resources to a role of "technology transition agents" with the task of producibility capability improvement and associated technology adoption. The SISPI technology transition authority will directly support these program-designated transition agents with advice on the applicability and readiness of adoptable technologies and provide guidance on effective practices for introducing and institutionalizing these technologies within their acquisition programs.

3.1 VALIDATION, INTEGRATION, AND PRODUCTIZATION

The Transition theme of Validation, Integration, and Productization (T-VIP) is concerned with determining the applicability of research results to DoD problems and integrating current and emerging technologies into a practicable, product-quality whole. Although validation, integration, and productization are distinct elements of transitioning research results into adoptable technology, these elements may be applied repeatedly in varying order as appropriate to mature each targeted technology. Because of the interplay among these elements, the roadmap envisions them as having intertwined objectives and interdependent milestones.

3.1.1 Definitions

3.1.2 Objectives

- Provide access to unclassified DoD system artifacts that provide the context for understanding DoD acquisition and sustainment challenges as formulated under SISPI research themes.
- Establish the means to evaluate the applicability, adoptability, and effectiveness of producibility technology in addressing DoD acquisition and sustainment problems.

- Promote community-focused efforts to create product line frameworks and associated assets that support DoD acquisition priorities.
- Establish an infrastructure in which Producibility technology can be integrated and used with other (e.g., commercial) tools or practices to evaluate compatibility and effectiveness.
- Transform producibility technologies into productized forms sufficient to permit adoption by industry for DoD acquisition and sustainment programs.

3.1.3 Topics

1. Validation (V)
2. Integration (I)
3. Productization (P)

3.1.4 Notional Milestones

(PY1) T-VIP-1	architecture and procedures for a family of open heterogeneous environments defining a standard framework within which development tools can be demonstrated and evaluated through use on DoD sample problems []
(PY1) T-VIP-2	a repository of DoD sample problems and assistance for converting and importing problems into a form usable by a tool []
(PY2) T-VIP-3	procedures and guidance for documenting and enhancing the mechanisms by which tools and methods are able to be integrated into communicating, coordinated suites []
(PY2-10) T-VIP-4	VIP capabilities routinely maintained and enhanced in support of transitioning SISPI technology into use on DoD programs []
() T-VIP-5	tools/methods criteria for evaluating effectiveness to a purpose, characterizing in terms of scope of applicability, assumptions, capabilities, and limitations, and rating as to readiness for adoption by DoD programs []
() T-VIP-6	a venue within which tool/method vendors and DoD product developers can communicate concerning future needs and weaknesses of current offerings []
() T-VIP-7	materials and sources for education, training, and support of readiness-rated tools and methods []
() T-VIP-8	assistance to commercial/open-source product suppliers in targeting improvements (e.g., enhanced capabilities or integration mechanisms) that DoD programs would commit to adopt []

3.1.5 Elaboration

Research toward a producibility objective results in some realization of technology, typically as a method or tool. Each such method or tool comes with assumptions about its intended usage, in particular how use of that method or tool fits into an SiS engineering process. T-VIP tasks are the

means for iteratively evaluating and maturing a technology realization until it attains a form that is viable for use in acquisitions.

Validation is envisioned as occurring within an environment that approximates the circumstances under which SiS engineers work. A complete validation environment would be tailorable to match the capabilities and resources, including limitations, of a representative SiS production environment that a technology is targeting for its adoption. To be realistic, the SISPI anticipates identifying representative problems and complementary solution assets that an SiS engineer might encounter. The scope and capabilities that a particular technology is meant to address will determine tailoring of the environment needed to have it represent appropriate problem scope and assets.

The initial basis for evaluating a technology and its realization will be the criteria that the providing researcher has specified as the expected value of that technology in SiS product engineering. The validation environment must provide for the measuring of technology usage as it is applied to representative problems. Any discrepancy between expectations and measured experience will be noted for action, in the form of revisions to the technology, to its realization, or to expectations for its value. When through this process a technology has been shown to have value commensurate with stated expectations, it becomes a candidate for adoption. Actual adoption will require communicating the demonstrable value of the technology to potential adopters, typically following further work to enhance the fit of the technology into SiS engineering practices and to improve the engineering of its realization to meet users expectations for product quality.

Integration requires an understanding of the contexts within which a technology is to be used. This may be known from the beginning or it may vary depending on the specific circumstances of candidate adopters and require corresponding tailoring of the technology realization. Integration focuses first on ensuring that information supplied to and from the technology realization (as a method or tool) is consistent and compatible with the sources/uses of that information. Secondly, it focuses on ensuring that the technology realization does not provide redundant capabilities that are more properly within the scope of another method or tool within the expected engineering environment. Thirdly, it focuses on constraints that are imposed with use of the technology such as any unresolvable incompatibilities with other methods or tools that might otherwise be used or additional activities imposed on the adopter such as having to perform a preceding transformation of stored data needed in use of the technology. Regardless of any revisions to a technology or its realization for purposes of improved integration, a responsibility of the technology effort is to comprehensively define for adopters what adjustments or accommodations they must make to use the technology, including a proper characterization of how the technology fits with current practices and technologies, and what assumptions the provider is making as to the practical limits within which the technology is useful.

Productization is concerned with ensuring that a technology realization is sound, usable, and free of error. A fundamental responsibility of the provider is either to ensure that a realization fully and correctly expresses the intended technology or to clearly communicate any limitations that have been necessary to impose. The provider must ensure that the means of realization are sound and consistent with the assumptions that characterize applicability of the technology. If the technology is realized in part or whole as a tool, the tool must be built following sound engineering practices in such a way that it is safely and correctly usable within the adopters' environment, that its usage and behavior are clearly and completely documented for adopters, that its observable behavior is consistent and within the bounds of the technology being realized, and that it has no

undocumented behaviors or effects. Evidence that these objectives have been met must be provided to the SISPI transition authority as an aid to promoting Producibility technologies to potential adopters. If post-production support such as user training and assistance or product maintenance and evolution are included as part of the productization effort, the conditions and mechanisms of such support and alternatives must be clearly communicated as factors of importance to potential adopters.

3.1.6 References

3.2 ADOPTION

The Transition theme of Adoption (T-A) is concerned with moving productized technology realizations into practice on DoD acquisition and sustainment efforts.

3.2.1 Definitions

ATTO	acquisition program TT organization
TT	technology transition
TTA	SISPI Technology Transition Authority

3.2.2 Objectives

- Identify productized technology realizations that provide producibility improvements in SiS engineering.
- Foster DoD policies and practices that accommodate producibility improvements.
- Identify opportunities for adoption of emerging producibility technologies.
- Assist DoD ATTO agents to formulate receptive SiS environments for the adoption of available producibility technologies.
- Monitor and evaluate uses of producibility technologies to influence future investment for greater benefit and applicability.

3.2.3 Topics

1. Acquisition policies and practices (A)
2. Technology transition authority (T)
3. Acquisition program TT organizations (P)
4. Domain-specific (product line) communities (C)

3.2.4 Notional Milestones

(PY3) T-A-A-1	acquisition guidance specifies use of semi-formal notations for the specification of system and software requirements [R-DM-R-1]
(PY1) T-A-A-2	acquisition guidance specifies that acquisition programs institute an ATTO including a principal methodology agent whose role is to evaluate and approve systems and software engineering and management processes, tools, and methods, with responsibility to report on plans and progress in instituting producibility improvements []

(PY4) T-A-A-3	acquisition policy requires that all acquired products be demonstrated prior to Production and Deployment in an approved producibility technology facility configured to represent the operational environment characterized in a concept of operation for the planned system []
(PY2) T-A-A-4	guidance to acquisition programs specifies that any system to be deployed in multiple versions be analyzed and engineered with a perspective of product line criteria []
(PY4) T-A-A-5	acquisition efforts are evaluated for effectiveness in identifying and managing uncertainty (opportunity and risk) and in accommodating known potential change in needs, environment/infrastructure, and technology []
(PY1) T-A-T-1	TTA established to identify and qualify adoptable producibility technologies []
(PY1) T-A-T-2	TTA offers resources that allow ATTO agents to maintain awareness of SISPI activities and the state of producibility technologies []
(PY3) T-A-P-1	ATTO agents report measurable activity-level improvements due to adopted producibility technology on a major DoD acquisition []
(PY5) T-A-P-2	ATTO agents report measurable program-level improvements due to adopted producibility technology on a major DoD acquisition []

3.2.5 Elaboration

Effective technology advances have no value unless DoD SiS acquisition programs adopt those advances into practice. This requires that these advances be productized as envisioned in the T-VIP theme and that government and industry practitioners understand their use and undertake associated required organizational actions to adopt them. (An example of such recommendations for the more limited case of software product lines illustrates the types of changes in acquisition practice that will be needed [Campbell 2002].) This is likely to happen effectively only if each acquisition program recognizes the need to charter an activity whose purpose is the active systematic selection and adoption of technology that benefits the effort. The Technology Development phase of the Acquisition Life Cycle envisions that each program will undertake this effort; previously, this has tended to focus on hardware manufacturing needs but there is a complementary need to address the technology needed for an appropriate level of SiS product manufacturing as conceived in the SISPI producibility vision.

3.2.6 References

4 Managing SISPI Efforts

SISPI governance will establish and manage a plan under which efforts addressing specific research and transition milestones can be proposed and prioritized. This will entail setting criteria for the prioritization of proposed efforts and establishing mechanisms for measurement-based evaluation of each funded effort's progress toward its objectives and of the overall effort's progress toward improving SiS acquisition and sustainment results.

4.1 DEFINITIONS

capability	the maximum production that can result in theory from the use of a specified configuration of business and technology practices
maturity	the degree to which an enterprise is effective in achieving a targeted level of capability in the performance of its business

4.2 OBJECTIVES

- Select among proposed technology research and transition efforts based on well-defined prioritization criteria.
- Determine whether technology innovations have the expected benefit within their scope of application.
- Determine whether individual acquisition programs are seeing benefits expected given the SISPI technologies they have adopted.
- Determine whether SISPI efforts are leading to improvements in the efficiency and effectiveness of the overall DoD acquisition-sustainment system.
- Refine SISPI plans to optimize benefits being experienced over all evaluation perspectives.

4.3 TOPICS

1. Prioritizing efforts (P)
2. Measuring progress (M)

4.4 NOTIONAL MILESTONES

(PY1) M-1 guidance for how each SISPI research proposal is to specify its expected impact in terms of goals and indicators for evaluating success []

(PY1) M-2 an enhanced set of SISPI-relevant metrics identified that are sufficient to characterize current practices within DoD acquisitions as a baseline for use in detecting future improvements []

(PY2) M-3 guidance to acquisition programs establishing metrics to be reported as a basis for detecting improvements correlated to any SISPI-related changes in practice [M-2]

(PY3) M-4 annual profile of the status of DoD acquisition-sustainment capabilities related to SISPI [M-3]

(PY3) M-5 a producibility-capability scale that categorizes degrees of production that can result due to the use of different technology practice configurations [M-1]

4.5 ELABORATION

In keeping with Deming [Deming 1986], "capability" here refers to the potential for production that a set of practices enables in an enterprise; "maturity" refers to the degree to which the enterprise is able to realize that potential. From this perspective, technologies can be evaluated in terms of whether they improve the capability of a given enterprise and transition can be focused on what efforts are needed to mature the use of those technologies, either by adjusting the technologies to fit the enterprise or by improving the ability of the enterprise to use them. SISPI governance will institute measurement efforts for the monitoring and evaluation of chartered research and transition efforts as to their effectiveness in improving the DoD enterprise of SiS realization.

4.5.1 Setting Measurable Goals for Research Efforts

For any proposed research topic, the submitter must define precise goals for themselves with measurable criteria for evaluating progress and success of associated research efforts that are to be funded. The submitter must define both the criteria and the measures that an adopter of the technology can make to determine that the technology is having its expected effects. These criteria will be used as a principle basis for determining when a technology is sufficiently mature to become the subject of transition. A submitter may change these criteria as research progresses but success will be viewed finally in terms of their being able to realistically characterize the value that a technology provides. The effectiveness of the SISPI as a whole will be judged on its being able to move producibility technologies from research into transition with reliable prediction of value to potential adopters.

4.5.2 Selecting SISPI Efforts

The SISPI will seek proposals for efforts targeting milestones identified in the technology roadmap. Each proposal will be expected to provide information useful in evaluating its potential value in light of SISPI objectives:

- Which roadmap milestone does the proposed effort address? What are the foundations and experience for this work that indicate its potential?
- What approach is proposed for achieving the milestone? What is the plan for a successful effort?
- What are the challenges and uncertainties that the effort must overcome to succeed? What other advances are needed to make use of this work?
- How does this approach fit with current practice? How will other practices be affected by adoption of this approach?

- What will be required to transition the results of this effort into practice? What benefits, at any level, is this expected to offer to practitioners who subsequently adopt it? What are the likely associated costs, both to adopt and to use? How can the benefits and costs be measured in practice?
- Based on the goals of this effort, what are its expected direct and indirect contributions to the producibility vision?

SISPI governance will evaluate each proposed effort in terms of criteria that weights its relative value against that of other proposed efforts. Among the criteria to be used will be the following:

- significance of the effort to achieving the producibility vision
- fit and timeliness of the effort for early use on SiS acquisition programs
- evidence for the technical feasibility of the effort
- compatibility of the approach with current/emerging DoD/industry practices and trends or dependencies on other advances
- identification of suitable measures for tracking the progress and success of the effort
- an argument for how successful results will translate into deployed technology, including prospects for productization and adoption

4.5.3 Opportunities for Near-Term Progress

The value of the SISPI will be greatest if it can orchestrate achievement of the SiS producibility vision but initial support requires that it also deliver early benefits to SiS acquisition programs.

As a seed for identifying efforts that have near-term value, but are on a path to the vision, these are some of the goals for research and transition actions that could be relatively low in effort to accomplish early:

- Define a standardized framework for precisely identifying and measuring critical software-system properties (R-PSA-1).
- Create a DoD-wide repository exhibiting large-scale use of effective software methods for requirements specification, architectural and component design, and verification (T-VIP-2).
- Reformulate relevant systems and software methods to foster collaborative software-based systems engineering.
- Institute federal/DoD acquisition practices that motivate programs to adopt practices that address common life-cycle software problems (T-A-A-5).
- Establish a DoD capability for facilitating the packaging and transition of effective R&D technology into use on acquisition programs (T-A-A-2).
- Initiate efforts on programs building multi-version solutions to create a model-driven development capability based on product line principles (T-A-A-4).
- Provide guidance on simple and safe techniques for building software that takes good advantage of multi-core and multi-processor computers.

4.5.4 Measuring SISPI Progress

Means are needed to ensure that SISPI efforts are yielding appropriate benefits. Benefits should be detectable at three levels:

1. technology-focused—within the immediate context in which a technology is applied, improving directly associated productivity or resulting work product quality
2. program-focused—over a single acquisition program that has adopted producibility technology, with the expectation that one or more of its measurable activities will exhibit productivity or product quality improvements
3. systemic—over the entirety of the DoD acquisition-sustainment system, reflecting the effects of multiple acquisition programs adopting various producibility enhancements that result in improved cost, quality, and timeliness in delivering capabilities into DoD operations

Technology-focused benefits are narrow in scope with results detectable in the near term (1-2 years after adoption). Program-focused benefits are somewhat more diffuse in scope, potentially involving multiple technology innovations and affecting different elements of a program, and consequently requiring longer to detect measurable results (2-4 years). Systemic benefits are the accumulation of the efforts of many programs attempting differing degrees of improvement using different mixes of technology, with effects being measurable only after substantial delay (4-8 years) and requiring statistical cost-benefit extrapolations from appropriate indicators of expected improvements.

In each of these categories, there may be multiple classes of stakeholder that are concerned with different measures of success. For example, engineers considering adopting technology may focus primarily on ease of use and product quality effects whereas an acquisition authority may focus more on effects on total product cost or on the availability of trained practitioners. SISPI measurement must properly address all such stakeholder perspectives.

Technology-Focused Measures

Technology research must, by its nature, focus on a limited scope of application. As such, there is no universal measure of its effectiveness in use. Rather, the advocate for any particular technology must specify what improvements should be expected through the use of that technology and what measures may be used to verify such improvements. As a rule, the use of a technology will often be specific to a single activity and its direct effect will be seen in measures associated with that activity and its work products. There may also be indirect effects of the use of a technology in related (influencing or dependent) activities, requiring those activities to be performed differently and resulting in effects on measures associated with those activities. More remote effects, particularly in later phases of a program's life cycle, may in some cases be relevant, such as when a technology's expected effects include reducing maintenance costs.

A technology's advocate must establish specific improvement goals that an adopter can expect to realize through the proper use of the technology. These goals will then be used by the SISPI to evaluate progress in research and to validate the readiness of the technology for adoption. Success in meeting these goals, which may be revised to ground expectations as research progresses, will be used by technology transition agents to evaluate applicability to particular programs' needs and to persuade programs to adopt appropriate technologies.

As a framework for specifying technology-focused measures, the advocate must define expected effects in terms of an assumed engineering life cycle and process in which the technology will be applied. SISPI provides the following canonical categorizations and the advocate may choose one of these or specify another to the same or greater granularity.

Life-cycle models and constituent phases

1. Linear/incremental/evolutionary (in accordance with DoDI 5000.2 Defense Acquisition Management Framework)
    a. Concept Refinement
    b. Technology Development
    c. System Development and Demonstration
    d. Production and Deployment
    e. Operations and Support
2. Iterative/spiral/agile
    a. Pre-acquisition
    b. Inception
    c. Elaboration
    d. Construction
    e. Transition
    f. Use and Maintenance/Evolution
3. Product line/mass customization
    a. Adoption/Improvement (Organizational Management)
    b. Domain Engineering (Core Asset Development)
        i. Conception
        ii. Elaboration
        iii. Expansion
        iv. Consolidation
    c. Application Engineering (Product Development)
    d. Application Use

Activities within phases of a process:

1. Project management
    a. Planning including budgeting, scheduling, and process and infrastructure definition
    b. Monitoring and reporting including measurement and review
    c. Coordination and control including tasking, risk management, quality assurance, and configuration management
2. System/software requirements definition

a. Concept of operations
b. Analysis of needs, capabilities, and opportunities (including variabilities analysis)
c. Requirements specification

3. System/software design
    a. Operational process definition
    b. System and software architectures (including data and concurrency)
    c. Design tradeoff analyses (including architecture and algorithms)
    d. Component interface specifications
4. System/software component development
    a. Component internal design
    b. Component implementation
    c. Component verification
5. System/software verification (test and evaluation including integration and developmental)
    a. Planning
    b. Preparation (environment and scenarios)
    c. Performance
    d. Analysis
6. System/software validation (test and evaluation including operational, user/acceptance, and assurance/certification)
    a. Planning
    b. Preparation (environment and scenarios)
    c. Performance
    d. Analysis
7. User support
    a. Documentation
    b. Training
    c. Assistance
8. System/software installation
    a. Hardware/software installation
    b. Data preparation
    c. Organizational transition

The technology advocate is asked to specify the anticipated effects of introducing their technology into an adopter's life cycle and process. The particular phases and activities that will be affected,

directly or indirectly, by the technology must be specified in terms of how appropriate measures of the indicated activities are expected to be affected. The SISPI will evaluate the maturity and need for improvement in the technology based on these identified measures, recognizing that these measures may change due to insights the advocate gains as a result of research and transition activities.

Program-Focused Measures

The starting point for program-focused measurement of SISPI effectiveness will be the metrics that programs report in accordance with DoD 5000.4. These consist of the following:

- project size
- project schedule
- development effort
- quality

DoD 5000.4 defines these measures loosely without precision, rather programs are given flexibility to use whatever indicators of these aspects that they consider applicable and, while required to report how measures are obtained, to measure them as they see fit. As a result, it is generally not valid to compare metrics obtained from different programs. Such comparisons are even more problematic from an SISPI perspective in that different programs may adopt different configurations of technology resulting in different expectations for improvement. However, the utility of these metrics for SISPI purposes hinges only on their being accurate indicators of the effects that producibility technology improvements may produce for a particular program. As such, the approach to be used will be to compare actual results to results that each program uses to estimate these measures. The expectation is that SISPI should be judged as effective only if programs are adopting technologies that result in measures that are significantly better than estimations based on past experience would predict. To be qualified as part of the basis for evaluating SISPI effectiveness, programs must identify the specific producibility technologies they have adopted and report these metrics both as estimated and as measured.

Systemic Measures

The essential systemic metrics for DoD acquisition and sustainment are long established, as identified in DoDD 5000.1:

- **timeliness:** time from identified need to productive use
- **affordability:** cost to develop, to manufacture, to sustain
- **effectiveness:** quality of operational capabilities

Although the DoD may view these metrics as desired attributes of individual acquisition and sustainment efforts, it is more important for SISPI to view these as the desired attributes of the entire system of DoD acquisition and sustainment. There are two aspects to these metrics: the degree to which the associated measures become more predictable and the degree to which these measures improve, as producibility technologies are adopted by acquisition programs. For example, timeliness is achieved if a program is able to adhere to a sound schedule but also if the absolute time from conception to productive use is reduced. In short, there is the need to demonstrate the ability not only to perform to expectations but also to meet more demanding expectations as advances occur.

Beyond determining that these metrics are improved as programs adopt producibility technologies, SISPI needs to collect data on expectations for these measures (initial and subsequent estimations) versus actual values, reasoning that producibility improvements will lead to results that better match initial expectations due to both more realistic expectations and greater abilities to understand and perform to expectations.

Bibliography

General References

These references represent principal sources of influence on the approach being proposed here for an SISPI:

Campbell, G. "Domain-Specific Engineering." Embedded Systems Conference, July 1997.

Software Design and Productivity Coordinating Group of the Interagency Working Group on Information Technology Research and Development. "Workshop on New Visions for Software Design and Productivity: Research and Applications." 2001.

High Confidence Software and Systems Coordinating Group of the Interagency Working Group on Information Technology Research and Development. *High Confidence Software and Systems Research Needs*. 2001.

United States General Accounting Office. *Defense Acquisitions—Stronger Management Practices are Needed to Improve DOD's Software-Intensive Weapons Acquisitions*, (GAO-04-393). Washington, DC: 2004.

DUSD-S&T. *Report on the 2005 Software Producibility Workshops (Draft)*. Washington, DC: 2005.

Basili, V. *Large Scale Information Systems and Software.* NIT Subcommittee of PCAST. 2006.

Broy, M. "The 'Grand Challenge' in Informatics: Engineering Software-Intensive Systems." *IEEE Computer 39* (10), October 2006.

Cited References

[ARTEMIS 2005]
ARTEMIS. *ARTEMIS Strategic Research Agenda*. 2005.

[Campbell 2002]
Campbell, Grady. *A Software Product Line Vision for Defense Acquisition* (CMU/SEI-2002-TN-002). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 2002. http://www.sei.cmu.edu/publications/documents/02.reports/02tn002.html

[Deming 1986]
Deming, W. E. *Out of the Crisis.* Boston, Mass.: MIT Center for Advanced Engineering Study, 1986.

[Forrester 2006]
Forrester, Eileen (ed.). *A Process Research Framework.* Pittsburgh, Pa.: The International Process Research Consortium, 2006.

[ITEA 2005]
ITEA Office Association. *ITEA2—European Leadership in Software-intensive Systems and Services*. 2005.

[Northrop 2006]
Northrop, Linda, et. al. *Ultra-Large-Scale Systems.* Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 2006. http://www-.sei.cmu.edu/publications/books/engineering/uls.html

[O'Connor 1995]
O'Connor, J., et al. "Reuse in Command-and-Control Systems." *IEEE Software 11*, 5 (September 1994): 73.

[Rajkumar 2007]
Rajkumar, R., et al. *Cyber-Physical Systems—A Research Agenda for National Competitiveness.* 2007.

[Turner 2006]
Turner, R., et al. *SISPI Cost/Benefit Return on Investment Analysis.* September 2006.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE December 2007	3. REPORT TYPE AND DATES COVERED Final
4. TITLE AND SUBTITLE Software-Intensive Systems Producibility: A Vision and Roadmap (v 0.1)		5. FUNDING NUMBERS FA8721-05-C-0003
6. AUTHOR(S) Grady Campbell		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213		8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2007-TN-017
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116		10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES		
12A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS		12B DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)

A Software-Intensive Systems Producibility Initiative (http://www.sei.cmu.edu/sispi) has been proposed to foster a program of technology research and transition that will improve producibility in the acquisition/development and sustainment/evolution of software-intensive systems (SiS).

This document is a draft in progress of a technology vision and roadmap to improve the ability of DoD and industry to deliver needed SiS capability in a timely, cost-effective, and predictable manner. The goal at this stage is to establish the general concepts and approach for a producibility initiative and to stimulate discussion of these ideas and the research and transition efforts needed to achieve enhanced producibility in practice.

The roadmap is meant to serve as a coherent evolving framework for defining and prioritizing potential research investments and technology transition efforts related to producibility. A roadmap has three elements: a representation of the current situation, a vision that characterizes an improved situation, and a plan of action for transitioning from the current to the improved situation. This roadmap identifies five research themes, two transition themes, and an approach to measuring effectiveness for an initiative focused on achieving a vision of enhanced SiS producibility.

14. SUBJECT TERMS producibility, software-intensive systems, acquisition	15. NUMBER OF PAGES 61
16. PRICE CODE	

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. Z39-18
298-102